L. 2.500
Frs. 3,75

BEST-SELLER MONDIALE

LA GRANDE ENCICLOPEDIA DI INFORMATICA PER RAGAZZI

10

IN SOLI 30 FASCICOLI



GRUPPO EDITORIALE JACKSON

IN COLLABORAZIONE CON AMSTRAD

Direttore responsabile
Paolo Reina

Direttore di divisione:
Roberto Pancaldi

Autori:
Judy Tatchell,
Nick Cutler,
Lisa Watts,
Mike Wharton,
Tony Potter,
Ivor Guild,
Ian Graham,
Lynn Myring,
Helen Davies,
Mike Wharton,
Ian Graham,
Brian Reffin Smith,
Lisa Watts,
Bill Bennett,
Judy Tatchell,
Jenny Tyler,
Lee Howarth,
Judy Tatchell,
Gaby Waters,
Graham Round,
Nick Cutler,
Gaby Waters,
Brian Reffin Smith,
Judy Tatchell,
Lee Howarth,
Cherry Evans,
Lee Howarth

Revisione e adattamento:
Martino Sangiorgio

Coordinamento editoriale:
Renata Rossi

Progetto grafico:
Sergio Mazzali

Distribuzione:
SODIP - Milano

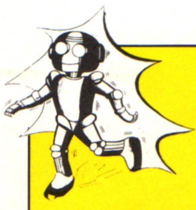
Stampa:
Vela - WEB - Vigano di Gaggiano (MI)

Direzione e Redazione:
Via Rosellini, 12 - Milano (20124) - Tel. 02/6880951 (5 linee)

Autorizzazione alla pubblicazione: richiesta
Spedizione in abbonamento postale Gruppo II/70
(autorizzazione della Direzione Provinciale delle PPTT di Milano)
Prezzo del fascicolo L. 2.500
I numeri arretrati saranno disponibili per 1 anno dal completamento dell'opera e potranno essere richiesti direttamente all'Editore a L. 3.000 (sovrapprezzo di L. 10.000 per spese d'imballo e spedizione).
I versamenti vanno indirizzati a:

Gruppo Editoriale Jackson S.p.A.
Via Rosellini, 12 - 20124 Milano

mediante emissione di assegno bancario oppure utilizzando il
C.C. Postale N. 11666203.
Non vengono effettuate spedizioni in contrassegno.



NEL PROSSIMO NUMERO:

- GIOCHI CON DATA
- USIAMO I VETTORI
- PROGRAMMA PER L'OROSCOPO
- TELESOFTWARE
- SUGGERIMENTI PER COSTRUIRE CIRCUITI
- I TERMINI DEL CHIP
- GIOCO ADVENTURE: NELL'INTERNO DEL PROGRAMMA

Esercizi con INKEY$

La funzione INKEY$ dice al computer di controllare la tastiera e vedere se è stato premuto qualche tasto, ma a differenza di INPUT, il computer non si ferma ad aspettarvi; il programma continuerà a girare. Alcuni computer usano una parola diversa da INKEY$, quindi controllate prima il comando per il vostro computer e poi provate gli esercizi di questa pagina.



Queste sono alcune parole usate al posto di INKEY$ nei vari computer. Consultate il vostro manuale o la Tabella di Conversione per il comando che il vostro computer accetta.

```
10 LET A$=INKEY$
20 IF A$="" THEN PRINT "!";
30 IF A$<>"" THEN PRINT A$;
40 GOTO 10
```

Provate a far girare questo programma usando il comando corretto per il vostro computer. Quando premete un tasto il computer stampa quel carattere, altrimenti stampa un punto esclamativo.



Adesso provate a scrivere un programma per far scrivere al computer la parola CIAO quando premete un tasto oppure lasciare uno spazio come illustrato nello schermo sopra.

Far aspettare il computer

```
10 LET A$=INKEY$
20 IF A$="" THEN PRINT " ";
30 IF A$<>"" THEN PRINT A$;
40 GOTO 10
```

```
10 LET N=0
20 LET A$=INKEY$
30 IF A$<>"" THEN GOTO 70
40 LET N=N+1
50 IF N<50 THEN GOTO 20
60 PRINT " ";:GOTO 10
70 PRINT A$;:GOTO 10
```



A volte può essere utile far aspettare il computer per un po' prima di proseguire con il programma. In alcune versioni di BASIC bisogna mettere un numero tra parentesi dopo INKEY$ come mostrato in alto a sinistra. Questo dice al computer quanto aspettare (in frazioni di secondo) prima di continuare. Se il vostro computer non vi consente di fare questo, potete ugualmente farlo aspettare inserendo INKEY$ in un loop e usando GOTO, come mostrato in alto a destra.

Errori di matematica ad alta velocità



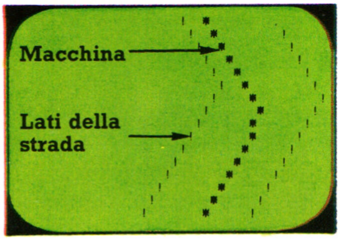
Riuscite ad individuare gli errori di questo programma e a correggerli per far girare il programma correttamente? Il computer dovrebbe scegliere due numeri a caso compresi tra 1 e 25. Voi dovete sommarli e premere un tasto qualunque non appena la risposta esatta appare sullo schermo.

```
10 CLS
20 PRINT "PREMI QUALSIASI TASTO QUANDO
30 PRINT "VEDI IL RISULTATO ESATTO
   DELLA SOMMA"
40 LET N=0
50 LET X=INT(RND(1)+25+1)
60 LET Y=INT(RND(1)*25+1)
70 PRINT
80 PRINT "X;" + ";Y;" = "
90 LET N=N+1
100 PRINT N
110 LET A$=INKEY$
120 IF A$<>"" THEN 180
130 FOR K=1 TO 400:NEXT K
140 IF N<30 THEN 90
150 PRINT "CHE SFORTUNA LA RISPOSTA ESAT
    TA E' ";X+Y
160 FOR K=1 TO 1000:NEXT T
170 GOTO 30
180 IF N<>X+Y THEN 150
190 PRINT "SI. LA RISPOSTA ESATTA E'
    X+Y"
```

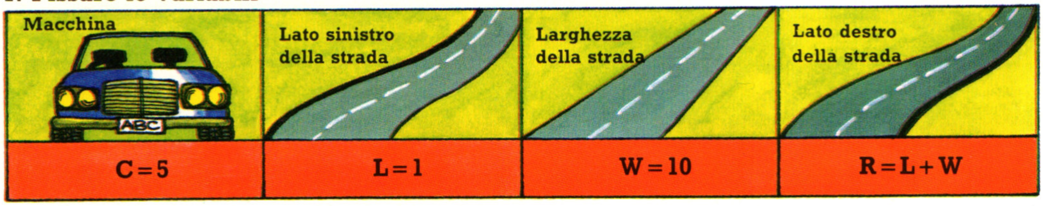


Come scrivere un gioco "car crash"

Seguendo i punti sottoelencati, provate a scrivere il programma per "car crash". Lo schermo a destra mostra come il gioco dovrebbe apparire. Per disegnare la macchina e la strada sullo schermo usate PRINT TAB con un asterisco (*) per la macchina e un punto esclamativo (!) per i lati della strada. Questa scorre a zig-zag lungo lo schermo mentre guidate con due tasti per evitare di schiantarvi sui bordi della strada.

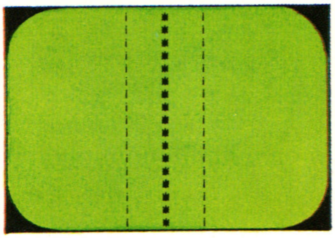


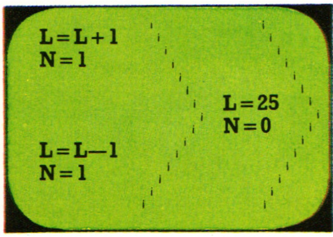
1. Fissare le variabili



Avete bisogno di quattro variabili C, L, W e R per definire le posizioni TAB per la macchina ed i margini della strada. Potreste aver bisogno di cambiare i numeri di sopra per riempire il vostro schermo. Pulite lo schermo e definite queste variabili nelle prime cinque righe.

2. Disegnare la strada



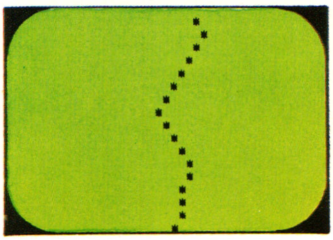


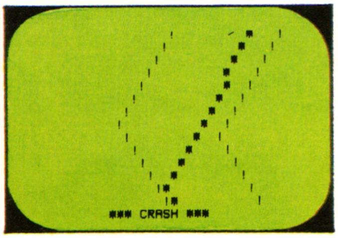


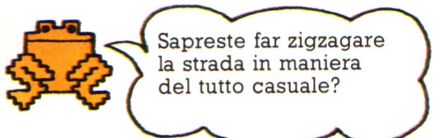
Adesso fate in modo che il computer disegni la strada e la macchina usando PRINT TAB con una variabile ed un simbolo. Usate GOTO per ripetere le istruzioni e disegnare una lunga strada diritta.

Per far zigzagare la strada dovete inserire alcune righe per cambiare il valore di L ogni volta che viene ripetuta l'istruzione PRINT TAB. Dovete pure evitare che la strada esca fuori dallo schermo. Per fare questo vi serve un'altra variabile (N). Ponete questa uguale a 1 se L < =1 e uguale a 0 se L > =alla larghezza dello schermo. Dite poi al computer di aggiungere o sottrarre 1 da L in funzione del valore di N.

3. Guidare la macchina







Per guidare la macchina avete bisogno di una riga con GET A$. Scegliete poi due tasti (< e >) e sommate o sottraete 1 da C in funzione di quale tasto venga premuto. Se il programma gira troppo velocemente per poter guidare la macchina, aggiungete un loop di ritardo.

Infine, dovete controllare se la macchina ha investito i margini della strada confrontando C con L e R. Quando si verifica un incidente, dovete segnalarlo al giocatore e far ripartire il gioco.



SOLUZIONI

Programma di un uomo che salta (pag. 67)

```
10 FOR J=1 TO 25 STEP 2
20 CLS
30 PRINT
40 PRINT TAB(J);" 0 "
50 PRINT TAB(J);"<O>"
60 PRINT TAB(J);" H "
70 FOR K=1 TO 200
80 NEXT K
90 CLS
100 PRINT TAB(J+1);"<0>"
110 PRINT TAB(J+1);" O "
120 PRINT TAB(J+1);" H "
130 FOR K=1 TO 200
140 NEXT K
150 NEXT J
```

Dovreste far girare il ciclo compreso tra le righe 10-150 tante volte quanti sono i caratteri che costituiscono il vostro schermo. Le righe 40-60 disegnano l'uomo nella posizione iniziale. Le righe 70-80 e 130-140 sono cicli di ritardo che fanno rimanere per un momento sullo schermo il disegno dell'uomo. Le righe 100-120 disegnano l'uomo nella seconda posizione.

Esercizi con IF/THEN (pag. 68, 82)

Prova della tabella pitagorica

```
55 IF B<>J*A THEN PRINT "SBAGLIATO. ";J;
" X ";A;" = ";J*A
```

Aggiungete questa riga per far dire al computer quando la vostra risposta è sbagliata.

Parola d'ordine (Password)

Aggiungete queste righe (con il vostro messaggio) per completare il programma della parola d'ordine.

```
10 LET S$="SALSICCE"
20 LET N=007
30 PRINT "PAROLA D'ORDINE PREGO";
40 INPUT P$
50 PRINT "NUMERO SEGRETO";
60 INPUT SN
70 IF P$<>S$ OR SN<>N THEN PRINT "SBAGLI
ATO. TESTA DURA"
80 IF P$=S$ AND SN=N THEN PRINT "O.K. CO
NTINUA"
```

Modificate il programma in questo modo affinché il computer vi chieda anche il numero segreto.

Il computer calcolatore

```
10 PRINT "PENSATE UN NUMERO ";
20 INPUT X
30 PRINT "ED UN ALTRO ";
40 INPUT Y
50 PRINT "VOLETE :"
60 PRINT "SOMMARE, SOTTRARRE,"
70 PRINT "DIVIDERE O MOLTIPLICARE ?"
80 INPUT A$
90 IF A$="SOMMARE" THEN LET A=X+Y
100 IF A$="SOTTRARRE" THEN LET A=X-Y
110 IF A$="DIVIDERE" THEN LET A=X/Y
120 IF A$="MOLTIPLICARE" THEN LET A=X*Y
130 PRINT "LA RISPOSTA E' ";A
```

Il computer esegue una certa operazione in funzione della risposta da voi data in A$.

Giochi degli indovinelli

```
45 IF Y<X THEN PRINT "TROPPO PICCOLO"
46 IF Y>X THEN PRINT "TROPPO GRANDE"
```

Aggiungete queste righe per far dire al computer quando il numero da voi pensato è troppo grande o troppo piccolo.

```
5 LET N=1
55 LET N=N+1
56 IF N>5 THEN STOP
```

Aggiungete queste righe per contare il numero di tentativi e fermare il programma dopo cinque tentativi.

Gioco delle parole

```
10 PRINT "LA PAROLA PREGO"
20 INPUT W$
30 PRINT "LA TRACCIA PREGO"
40 INPUT C$
50 CLS
60 PRINT "TRACCIA :"
70 PRINT C$
80 PRINT "PENSA LA PAROLA"
90 INPUT G$
100 IF G$=W$ THEN GOTO 130
110 PRINT "NO"
120 GOTO 90
130 PRINT "SI"
```

Le righe 10-40 fanno sí che il computer vi chieda una parola ed una traccia.
Le righe 50-60 puliscono lo schermo e stampano il suggerimento. La riga 120 indirizza il computer alla riga 90 per un altro tentativo.

Corsa dei cavalli

Ecco le righe IF/THEN complete per il gioco della corsa dei cavalli.

```
80 IF G1=H1 AND G2=H2 THEN GOTO 160
90 IF G1=H2 OR G2=H1 THEN GOTO 150
100 IF (G1=H1 AND G2<>H2) OR (G2=H2 AND
G1<>H1) THEN GOTO 140
120 IF N=4 THEN GOTO 170
```

Riga 80: Inviate il computer alla riga 160 se entrambe le vostre ipotesi sono esatte.
Riga 90: Se il cavallo è esatto ma il posto è sbagliato, inviate il computer alla riga 150.
Riga 100: Se una sola ipotesi è corretta andate alla riga 140.
Riga 120: Dopo quattro tentativi indirizzate il computer alla riga 170.

Idee per migliorare il gioco

```
15 LET S=0
140 PRINT "UNA PREVISIONE CORRETTA":LET
S=2:GOTO 120
160 PRINT "ESATTO":LET S=4
180 PRINT "IL VOSTRO PUNTEGGIO E' ";S
```

Ecco un'idea per un sistema di punteggio che dà quattro punti per i due cavalli indovinati e due punti per un cavallo esatto. La variabile S ricorda il punteggio.

```
190 PRINT "VOLETE GIOCARE ANCORA ? (S/N)
"
200 INPUT Z$
210 IF Z$="S" THEN GOTO 10
220 IF Z$="N" THEN STOP
```

Perché il giocatore possa girare ancora il programma, aggiungete queste righe.

Numeri casuali (pag. 100, 101)

Numeri casuali tra 10 e 20

```
INT(RND(1)*11+10)
```

Per fare questo moltiplicate per il numero delle cifre comprese nell'intervallo (11) e poi aggiungete il primo numero del range (10). usate il comando RND del vostro computer.

Un programma per un gioco

```
20 LET X=INT(RND(1)*20)+1
```

Modificate la riga 20 del gioco degli indovinelli per far scegliere al computer un numero tra 1 e 20.

Sequenza di numeri

Aggiungete queste righe al programma per il gioco della sequenza dei numeri.

```
5 PRINT "INDOVINA IL NUMERO"
7 PRINT "SUCCESSIVO DI"
8 PRINT "QUESTA SERIE"
10 LET X=INT(RND(1)*10+1)
20 LET Y=INT(RND(1)*10+1)

60 LET A=X+4*Y
70 INPUT N
80 IF N=A THEN PRINT "ESATTO"
90 IF N<>A THEN PRINT "SBAGLIATO. E'";A
100 GOTO 5
```

La riga 60 elabora l'ultimo numero della serie.

Sequenza a caso

```
24 LET R=INT(RND(1)*3+1)
25 IF R=1 THEN GOTO 30
26 IF R=2 THEN GOTO 40
27 IF R=3 THEN GOTO 50
```

Per selezionare una sequenza a caso, fate scegliere al computer un numero a caso tra 1 e 3 e memorizzatelo in R. Indirizzate il computer verso sequenze differenti in funzione del valore di R.

```
30 FOR I=1 TO 3
33 PRINT X+I*I
35 NEXT I
37 LET A=X+4*4
39 GOTO 70

40 FOR I=1 TO 3
43 PRINT I*I-Y
45 NEXT I
47 LET A=4*4-Y
49 GOTO 70

50 FOR I=1 TO 3
53 PRINT X+Y-I*I
55 NEXT I
57 LET A=X+Y-4*4
```

Per aggiungere al programma tre diverse sequenze di numeri, cancellate le righe 30-60 e aggiungete un ciclo separato per ogni sequenza. Alla fine delle prime due sequenze, indirizzate il computer alla riga 70.

Fuga da Zorgos

```
10 LET C=10
20 PRINT "VOI AVETE ";C;" CHIP"
30 PRINT "FATE LA VOSTRA SCOMMESSA :";
40 INPUT B
50 IF B>C THEN PRINT "VOI NON AVETE TUTT
I QUEI CHIP":GOTO 20
60 LET C=C-B
70 LET X=INT(RND(1)*6+1)
80 LET Y=INT(RND(1)*6+1)
90 PRINT "PREMETE P PER LANCIARE:";
100 INPUT P$
110 IF P$<>"P" THEN GOTO 90
120 PRINT TAB(5);X;TAB(10);Y
130 IF X=Y THEN GOTO 250
140 IF X+Y=10 OR X+Y=11 THEN GOTO 210
150 IF X+Y=6 OR X+Y=7 THEN GOTO 190

160 PRINT "SPIACENTE. AVETE PERSO LA SCO
MMESSA"
170 IF C=0 THEN GOTO 290
180 GOTO 20

190 PRINT "NESSUN RISCHIO. MANTENETE LA
PUNTATA"

200 LET C=C+B: GOTO 20

210 PRINT "O.K. TRIPLICATE LA PUNTATA"
220 LET C=C+(B*3)
230 IF C>=50 THEN GOTO 320
240 GOTO 20

250 PRINT "RADDOPPIATE LA PUNTATA"
260 LET C=C+(B*2)
270 IF C>=50 THEN GOTO 320
280 GOTO 20

290 PRINT "PECCATO. NON AVETE PIU' CHIP"

300 PRINT "NON POTRETE SCAPPARE DA ZORGO
S"
310 STOP
320 PRINT "BENE. ADESSO POTETE SCAPPARE"

330 PRINT "DA ZORGOS CON ";C;" CHIP"
```

- Riga 10: C conta il numero dei chip.
- Righe 50: Accertatevi che la scommessa del giocatore non superi il numero dei chip.
- Riga 60: Conta i chip rimasti.
- Righe 70-80: Sceglie due numeri a caso tra 1 e 6.
- Riga 110: Controlla l'input del giocatore.
- Riga 120: Stampa i numeri a caso sullo schermo.
- Righe 130-150: Controlla i numeri e indirizza il computer verso le diverse righe per elaborare i vari punteggi.
- Riga 170: Controlla il numero dei chip rimasti. Se C=0 invia il computer alla fine del programma.
- Riga 180: Rimanda all'inizio per un'altra scommessa
- Riga 200: Corregge il valore di C.
- Riga 230: Se C è maggiore o uguale a 50, indirizzate il computer alla fine del programma.

Carta, Pietra o Forbici: Individuate gli errori

Correggete le seguenti righe, come sotto mostrato, per far girare il programma correttamente.

```
50 LET R=INT(RND(1)*3+1)

80 IF R=3 THEN LET C$="FORBICI"

150 IF C$="FORBICI" AND A$="PIETRA" THEN
 LET F=1

190 PRINT "QUINDI ";

210 IF F=1 THEN PRINT "VINCI TU"

230 IF F=0 THEN LET C=C+1
240 IF F=1 THEN LET A=A+1

280 IF C<10 AND A<10 THEN GOTO 40
```

Elaboriamo i caratteri (pag. 113-116)

Giochi di stringhe

1

```
40 PRINT TAB(K);MID$(N$,K,1)
```

Fa stampare una lettera per volta della parola, alla posizione K di TAB, iniziando da quello numero K.

2

```
5 PRINT "UNA PAROLA PREGO"
10 INPUT N$
20 LET L=LEN(N$)
30 FOR J=L TO 1 STEP -1
40 PRINT MID$(N$,J,1);
50 NEXT J
```

Usate un ciclo con step —1 per far stampare la parola all'incontrario.

3

```
50 PRINT RIGHT$(S$,L-J)+LEFT$(S$,J)
```

Ogni volta che il ciclo si ripete, RIGHT$(S$,L-J) fa stampare al computer L (lunghezza parola) meno J lettere dalla destra di QUADRATO. LEFT$ aggiunge J lettere dalla sinistra. Provate con diverse parole.

La parola piú lunga

```
50 IF LEN(W$)>LEN(A$) THEN LET A$=W$
```

Ogni volta che il ciclo si ripete il computer confronta la lunghezza di W$ con quela di A$. Se W$ è maggiore di A$, esso sostituisce A$ con W$ e poi confronta la parola successiva. Alla fine del ciclo A$ conterrà la piú lunga parola da voi inserita.

La parola piú breve

```
10 LET A$="XXX!!!&&ABC***123!!!XXXXXXX"
50 IF LEN(W$)<LEN(A$) THEN LET A$=W$
70 PRINT "LA PAROLA PIU' BREVE :"
```

Programma simile a quello della parola piú lunga salvo le righe 10 e 50. La riga 10 contiene una stringa che il computer paragonerà con la lunghezza di ogni parola. Non conta quali caratteri metterete in A$.

Word editor

Ecco le righe che completano il programma per la correzione delle parole.

```
40 LET S$=" "+S$+" "
70 LET W$=" "+W$+" "
```

Avete bisogno di aggiungere degli spazi da entrambi i lati di S$ e di W$ per essere sicuri che alla riga 140 il computer ricerchi solo parole complete. Senza gli spazi il computer potrebbe scegliere qualsiasi carattere della frase che sia uguale a S$ o a W$.

```
100 LET LS=LEN(S$)
110 LET LW=LEN(W$)
```

Queste righe rendono LS uguale alla lunghezza di S$ e LW pari alla lunghezza di W$. La riga 100 viene ripetuta alla riga 160 per rendere LS pari alla lunghezza della nuova frase.

```
140 IF MID$(S$,K,LW)=W$ THEN LET A$=S$
```

Ogni volta che il ciclo si ripete, MID$(S$,K,LW) fa controllare LW (la lunghezza di W$) caratteri di S$, partendo dal carattere K. Quando il computer trova una sequenza di caratteri che uguaglia W$, esso immagazzina l'intera frase in una nuova variabile A$.

```
150 IF A$=S$ THEN LET S$=LEFT$(A$,K)+
N$+RIGHT$(A$,LS-(K+LW-2))
```

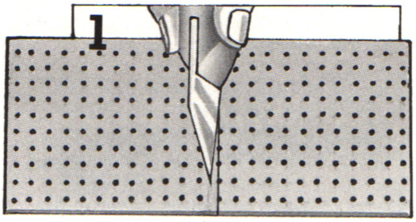
LEFT$(A$,K) elabora il numero di caratteri alla sinistra della parola che volete sostituire. Per capire come funziona questa riga provatela con una frase su un foglio di carta.

```
180 IF K<=LS-LW+1 THEN GOTO 140
```

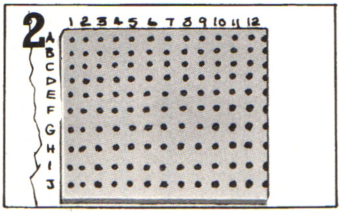
Rimandate il computer alla riga 140 per controllare il resto della frase nel caso che la parola che volete sostituire si presenti ancora una volta.



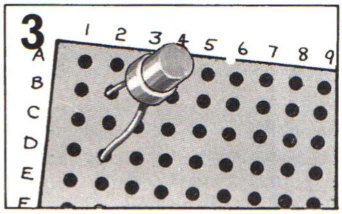
Costruzione del circuito indicatore



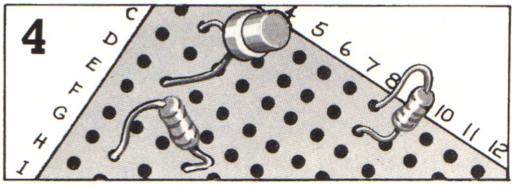
Spezza l'altra Veroboard in modo da avere un pezzo con 12 fori e 10 tracce; per spezzarla, incidila prima con un coltello affilato.



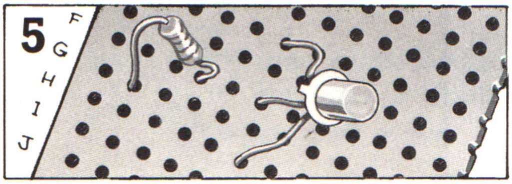
Poggia la scheda su un pezzo di carta con le tracce orizzontali sul retro e disegna un reticolo come quello della figura.



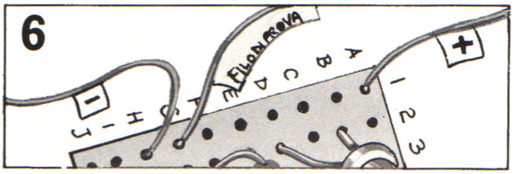
Salda il LED nei fori B2 e D2. Il filo accanto al bordo piatto (o quello piú spesso) deve andare nel foro D2.



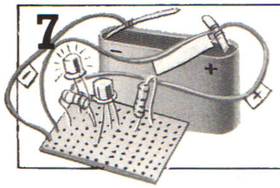
Salda la resistenza da 1K (strisce marrone, nera e rossa) in G2 e F4. Salda la resistenza di 270 ohm (strisce rossa, viola e marrone) in A8 e B8.



Metti il transistor con il collettore in D6, la base in F6 e l'emettitore in H6.



Salda un pezzo di filo lungo 30 cm nel foro A1 e contrassegnalo con "+"; saldane un altro in H1 e marcalo con "–", infine salda un filo di 6 cm in G1 e contrassegnalo "Filo di prova".



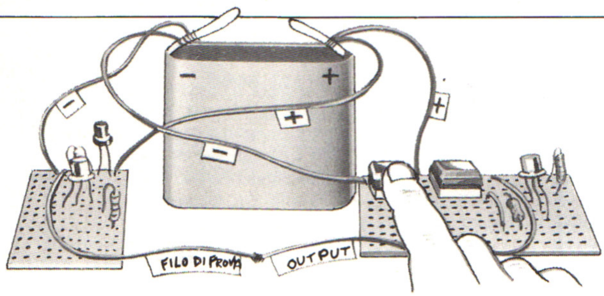
Se con il filo di prova tocchi il terminale negativo, il LED non si accende.

Per controllare se l'indicatore funziona, collegalo alla pila e tocca il terminale positivo della pila con il filo di prova: il LED dovrebbe accendersi mostrando che è presente un voltaggio elevato (segnale 1).*

Test della porta NOR

Per controllare una porta NOR, la si deve collegare al circuito indicatore. Unisci il filo di output della porta NOR e il filo di prova dell'indicatore; li puoi anche saldare, per ottenere una congiunzione piú solida.

Unisci poi il filo + della porta NOR e quello + dell'indicatore e collegali al terminale positivo della pila. Fai lo stesso con i due "–" e collegali al terminale negativo.



Appena i circuiti sono collegati, il LED verde dovrebbe illuminarsi, mostrando che non viene premuto nessuno dei due interruttori (una porta NOR dà un output di voltaggio elevato solo quando riceve due input di voltaggio basso). Se premi uno o entrambi gli interruttori, la luce si spenge, mostrando che l'input è diventato 1 e l'output 0.

* Se l'indicatore non funziona, controlla se tutte le componenti sono nei fori giusti e se i giunti sono saldati bene.

Ancora sulle porte logiche

I circuiti logici sono utili perché generano sempre lo stesso output per lo stesso insieme di input. Una porta logica con due input, per esempio una porta NOR, può ricevere quattro diversi schemi di segnali binari: 0,0; 0,1; 1,0; 1,1. Gli ingegneri e i progettisti di chip scrivono tavole di input e di output che mostrano tutti i risultati che una porta logica può produrre. Riportiamo qui sotto la tavola di una porta NOR.

INPUT A	INPUT B	OUTPUT
0	0	1
1	0	0
0	1	0
1	1	0

Tavola di verità di una porta NOR.

Se hai costruito la porta NOR delle pagine precedenti, puoi controllare la sua tavola di verità. Etichetta gli interruttori A e B e scorri la tavola riga dopo riga, premendo l'interruttore A tutte le volte che c'è un 1 all'input A e l'interruttore B quando c'è 1 in input B. Il LED verde mostra l'output.

Il nome di una tavola di input e di output è tavola di verità. Se vuoi puoi rileggere le descrizioni delle porte AND, OR e XOR e provare a scriverne le tavole di verità. Puoi controllare le tue risposte in seguito.

La storia delle porte logiche

I principi alla base delle porte logiche elettroniche e delle tavole di verità vennero concepiti verso il 1850 dal matematico irlandese George Boole, che studiava in che modo le persone arrivano a prendere una decisione. Mai avrebbe immaginato che le sue idee sarebbero state utilizzate per progettare circuiti elettronici.
Durante i suoi studi, si rese conto che nei processi decisionali sono implicati tre procedimenti logici principali, che chiamò AND, OR e NOT. Sono questi procedimenti che vengono imitati dalle porte logiche.
Boole inventò le tavole di verità per mostrare che, quando si è sicuri che un'affermazione è vera o falsa (e non una via di mezzo), è possibile prevedere tutte le conseguenze di un determinato procedimento mentale. Sotto vediamo la tavola di verità di Boole per un procedimento AND. Boole scrisse anche le tavole di verità per OR e NOT.

AFFERMAZIONE A	AFFERMAZIONE B	RISULTATO
Per es.: splende il sole. 	Per es.: indosso una pelliccia. 	Per es.: ho troppo caldo. 
FALSO	FALSO	FALSO
VERO	FALSO	FALSO
FALSO	VERO	FALSO
VERO	VERO	VERO

Se sostituisci la parola vero con il simbolo binario 1 e falso con 0, questa tavola di verità è esattamente uguale a quella di una porta AND elettronica, come puoi verificare più avanti.

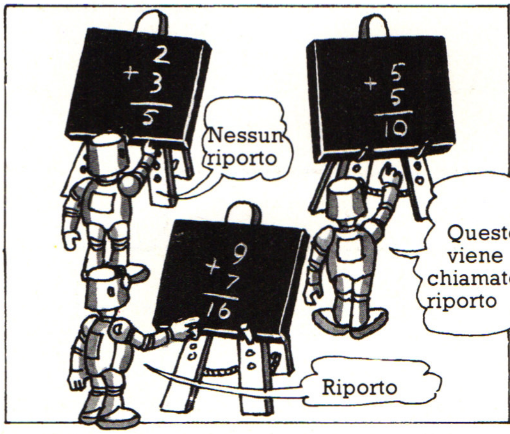
Quando gli uomini si sono resi conto di poter costruire circuiti elettronici in grado di fornire gli stessi risultati dei procedimenti mentali logici, i microprocessori e i computer sono diventati una realtà.

In che modo la ALU esegue le operazioni

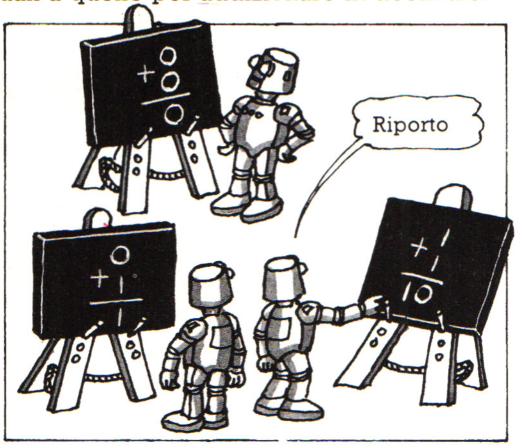
Raggruppando le porte logiche in un determinato modo, è possibile costruire un circuito che addizioni due numeri. Per capire il circuito, si devono conoscere le regole dell'addizione in binario, sotto riportate.
Il circuito di addizione può solo sommare una cifra binaria a un'altra; per addizionare codici binari a otto bit, la ALU richiede un gruppo di otto circuiti di addizione collegati fra loro, chiamato addizionatore.

L'addizione in binario

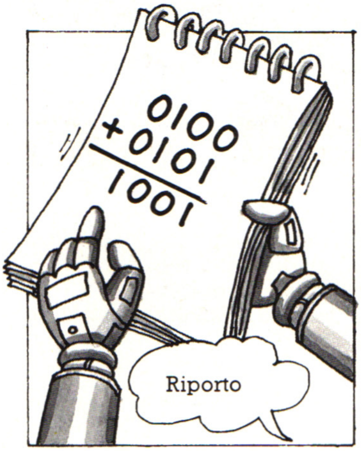
Le regole per addizionare in binario sono uguali a quelle per addizionare in decimale.



Nel sistema decimale la cifra piú grande è 9; quando la somma di due cifre è maggiore di 9 (per es. 5+5 o 9+7) la risposta viene mostrata riportando 1 e ricominciando da 0 nella colonna di destra.



Nel sistema binario le regole per addizionare sono uguali, però, poiché la cifra piú alta è 1, non appena si addiziona 1+1 è necessario il riporto. Come dall'esempio qui sopra.



Qui puoi vedere come vengono addizionati due numeri binari a quattro bit. Puoi controllare il risultato dell'operazione trasformando tutti in numeri in numeri decimali, come viene spiegato a destra.

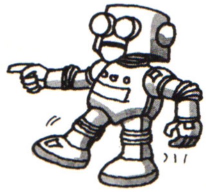
Trasformazione da binario in decimale

BINARIO	1		1	11
DECIMALE	8		2	14

0100 = 4

0101 = 4 + 1 = 5

1001 = 8 + 1 = 9



In un numero binario, ogni cifra ha un valore doppio rispetto a quello alla sua destra, quindi per trovare l'equivalente decimale dei numeri binari devi raddoppiare il numero decimale ogni volta che ti sposti a sinistra di un posto nel numero binario. Tramite la tabella qui sopra puoi vedere come il numero binario 0100 corrisponda a 4 decimale, 0101 sia 5 e 1001 sia 9 (4+5=9).

Progetto di un circuito di addizione

Nel progettare un circuito elettronico per addizionare due cifre binarie, è necessario costruire una tavola di verità che mostri tutte le addizioni possibili con i loro risultati, poi realizzare un circuito con la stessa tavola di verità.
Poiché il sistema numerico binario ha solo due cifre, le addizioni possibili sono solo quattro,* e vengono mostrate qui sotto. Per scriverle sotto forma di tavola di verità, le cifre da addizionare devono costituire l'input e la risposta l'output. E' inoltre necessario che gli output siano due, per tener conto anche del riporto.

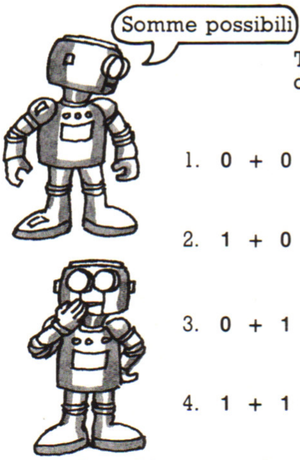
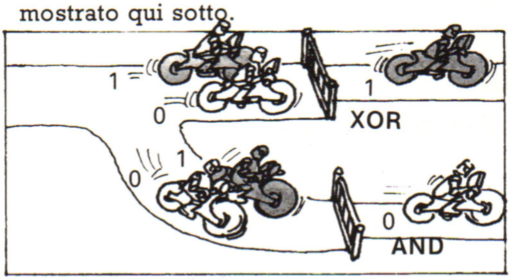


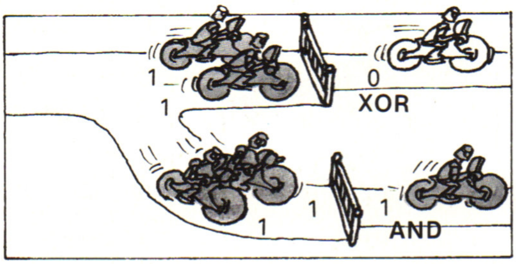
Tavola di verità

	INPUT PER LA PRIMA CIFRA BINARIA	INPUT PER LA SECONDA CIFRA BINARIA	OUTPUT PER LA COLONNA DI DESTRA	OUTPUT PER IL RIPORTO
1. 0 + 0 = 0	0	0	0	0
2. 1 + 0 = 0	1	0	1	0
3. 0 + 1 = 1	0	1	1	0
4. 1 + 1 = 10	1	1	0	1

Quali porte logiche utilizzare

Un circuito che generi la tavola di verità dell'addizione (mostrata qui sopra) può essere costruito utilizzando solo due porte logiche: una porta XOR darà l'output della colonna di destra del risultato e una porta AND darà l'output del riporto. (Puoi controllare guardando le tavole di verità di AND e XOR mostrate in seguito). Gli input, cioè le cifre da addizionare, devono essere inserite contemporaneamente a entrambe le porte, come mostrato qui sotto.





Queste figure fanno vedere come il circuito di addizione eseguirebbe 1+0 e 1+1. All'interno della ALU c'è un gruppo di otto circuiti come questi, ognuno dei quali addiziona un bit di un byte di un codice binario. C'è anche un secondo insieme di circuiti che addiziona il riporto di un circuito di addizione al circuito di addizione alla sua sinstra. Se l'ultimo bit di un byte dà un riporto, viene indicato dal flag di riporto nel registro di flag (vedi pagina 105).

Come vengono eseguiti gli altri calcoli

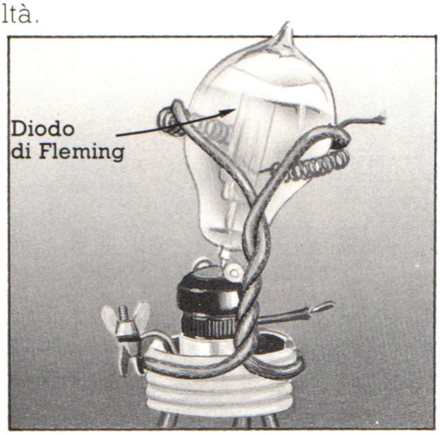
L'ALU esegue tutte le operazioni tramite addizioni nei circuiti di addizione. Per una sottrazione, uno dei due numeri viene reso negativo e poi aggiunto all'altro. La moltiplicazione viene effettuata continuando ad addizionare, e la divisione continuando a sottrarre. Anche i calcoli complessi, come l'estrazione di radici quadrate, vengono suddivisi in passaggi semplici, ognuno dei quali implica solo un'addizione. Un insieme di regole per eseguire calcoli complessi viene chiamato algoritmo; nel chip di una calcolatrice tascabile, i circuiti di ROM contengono un algoritmo per ogni operazione matematica che la calcolatrice può eseguire.

* Nel sistema numerico decimale, che ha dieci cifre diverse, le somme possibili addizionando due numeri sono cento.

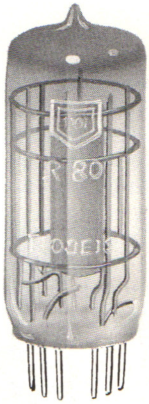
Storia del chip

L'invenzione del chip ha origine dallo sviluppo dell'elettronica (il controllo dell'elettricità) che nella prima metà del secolo è stato rapidissimo. Le prime componenti elettroniche sono state utilizzate per riprodurre il suono nelle radio e per controllare la velocità dei motori elettronici. Nelle prossime due pagine descriveremo la realizzazione delle primissime componenti elettroniche e come il chip sia divenuto una realtà.

Solo agli inizi del secolo gli scienziati si sono resi conto che era possibile controllare una corrente elettrica. Nacque cosí l'elettronica, nel 1904, con l'invenzione della valvola a diodo termoionico, o semplicemente diodo, da parte di Ambrose Fleming. Il diodo consentiva il flusso di corrente solo in una direzione e funzionava emettendo elettroni (particelle con una carica elettrica) da un filo riscaldato all'interno di un tubo di vetro vuoto. Gli elettroni venivano attratti da una piastra metallica all'altra estremità del tubo, cosí la corrente scorreva in quella direzione.

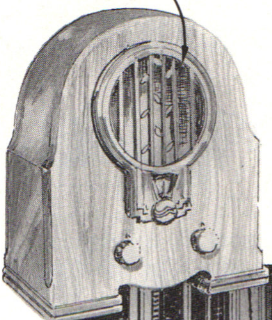


Triodo

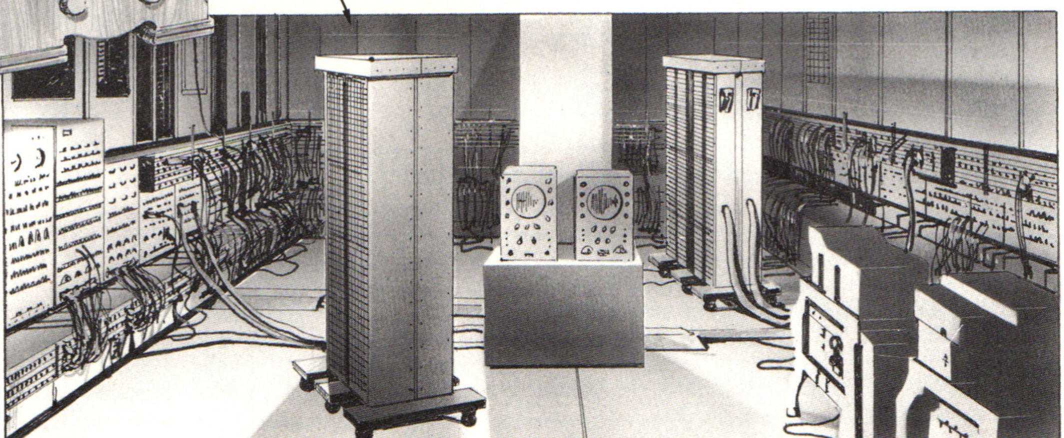


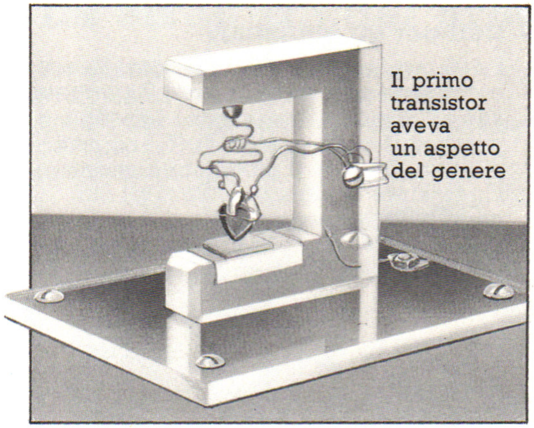
In quello stesso anno, Lee de Forest scoprí che inserendo una griglia metallica in mezzo al diodo si poteva controllare e modificare il flusso di corrente. Questa invenzione, chiamata triodo, fu particolarmente importante per lo sviluppo della radio e della televisione perché consentiva di amplificare (rendere piú grandi) le correnti. Venne anche scoperto che il triodo poteva funzionare da interruttore. Questo portò ai primi computer elettronici degli anni quaranta: Colossus in Inghilterra ed ENIAC negli Stati Uniti. ENIAC sta per *Electronic Numeric Integrator and Calculator* (calcolatore e integratore numerico elettronico). La macchina, aveva ben 18.000 valvole; riempiva una grande stanza e produceva un enorme quantità di calore. I tecnici dovevano lavorare quasi 24 ore al giorno per sostituire le valvole surriscaldate.

Una radio a valvole degli anni trenta



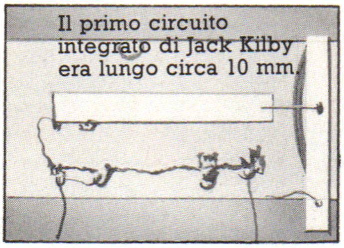
ENIAC

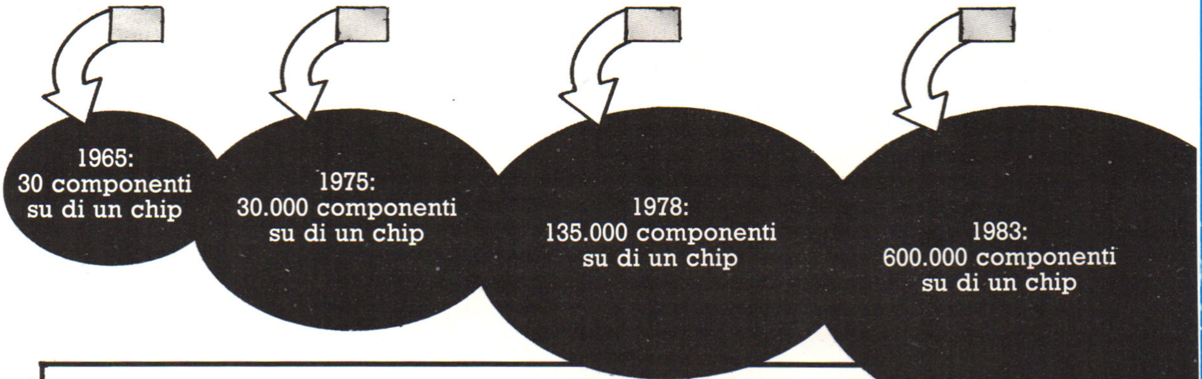




Nel 1947, tre scienziati americani, Bardeen, Brattain e Shockley, inventarono un dispositivo che divenne noto come transistor. Il transistor poteva svolgere le stesse operazioni di un triodo, ma era piú piccolo, solido e non si riscaldava. I primi transistor erano costruiti con il germanio, un semiconduttore; solo successivamente venne utilizzato il silicio.
Le prime applicazioni commerciali dei transistor furono negli apparecchi acustici, ma a poco a poco sostituirono le valvole nelle televisioni, registratori, radio e in apparecchiature molto piú grandi, come le centraline telefoniche. La prima volta che vennero usati nei computer fu nel 1953.

Nel 1958, Jack Kilby, della Texas Instruments, ebbe l'idea di riunire due transistor su un cristallo di silicio e costruí cosí il primo circuito integrato. Il bisogno di miniaturizzazione creato dai programmi spaziali e di difesa americani spinse gli scienziati a cercare nuovi metodi per far entrare sempre piú componenti in un chip di 5mm di lato. La figura qui sotto dà un'idea della velocità del progresso.





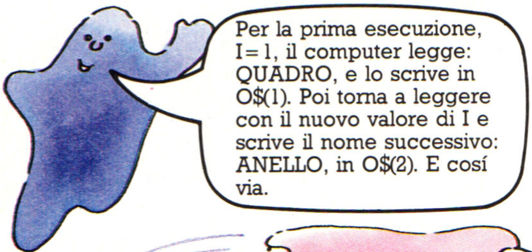
Il futuro

I costruttori cercano continuamente di rendere i chip piú potenti aumentando il numero delle loro componenti e la velocità di funzionamento. Attualmente la ricerca si accentra su un interruttore molto veloce e di bassa potenza che sostituirebbe i transistor. Il nuovo superinterruttore è la giunzione di Josephson, ideato nel 1962 da Brian Josephson, dell'università di Cambridge. Una giunzione di Josephson è costituita da due sottili strati di metallo separati da uno strato isolante piú sottile. A temperature di centinaia di gradi sotto zero, i metalli diventano "superconduttori" e le giunzioni si aprono e si chiudono dieci volte piú rapidamente dei piú veloci transistor attuali. Si pensa che usando una tecnologia basata sulla giunzione di Josephson, un computer delle dimensioni di una mela potrebbe elaborare tutte le informazioni attualmente gestite da un computer che riempie una stanza, e piú rapidamente. Dovrebbe lavorare a temperature molto basse.
Il successivo stadio di sviluppo di cui si parla è la produzione biologica dei chip.

Mettere i dati nelle matrici

Avendo predisposto le aree di immagazzinamento con i loro nomi nella memoria del computer, dovete ora dirgli che cosa scriverci. Un modo per far questo è quello di elencare i dati in ordine, ed ordinare al computer di leggerli ad uno ad uno, scrivendoli uno alla volta nelle caselle di una matrice. Ecco come si fa, con istruzioni BASIC:

```
DIM O$(W)
DATA IL QUADRO,L'ANELLO,INCANTESIMI
MAGICI,IL CALICE,ETC.
FOR I=1 TO W
READ O$(I)
NEXT I
```

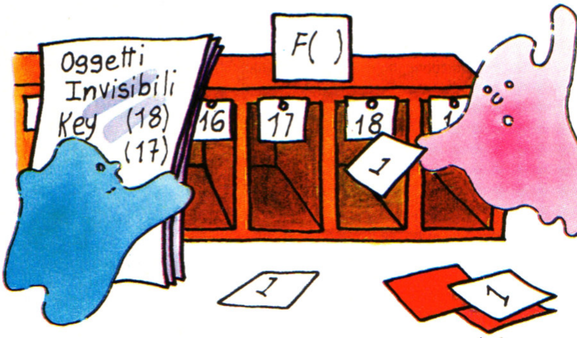




Questo è il ciclo di READ (lettura) dei termini DATA in O$( ). Guardate il listato del programma finale, alle linee 1600-2100, e cercate di individuare i cicli di lettura dati per le altre matrici.

Dati per gli indicatori

I dati per la matrice di indicatori F( ) consistono solo di uni e zeri. Gli oggetti che all'inizio del gioco sono invisibili prestano 1 nella casella di F( ) ad essi relativa. Quando il giocatore li scopre, l'indicatore cambia in 0. Tutti gli altri oggetti all'inizio sono contrassegnati con zeri. Voi dovrete quindi solo dire al computer quali caselle F( ) devono



contenere. 1. Lasciare le altre vuote è lo stesso che scrivervi zero. Il modo più semplice per riempire questa matrice è mostrato alla linea 2090 del programma finale. Potreste aver notato che certe caselle di F( ) non vengono usate perchè alcuni oggetti non cambiano di stato. Questi indicatori in più possono essere usati per altre cose. Per esempio, nella "Casa Stregata", la casella F(14) (indicatore della corda) è usata per indicare se il giocatore è salito sull'albero. La candela ha bisogno di due indicatori: uno per mostrare se è visibile, l'altro per mostrare se è accesa. L'indicatore F(0), non utilizzato, viene allora usato per "accendere" la candela. Se vi serve un indicatore in più per qualche cosa, potete utilizzare le caselle relative a parole che non ne usano, ad esempio la parola "NORD" (casella 19).

Dati per la matrice di oggetti posseduti

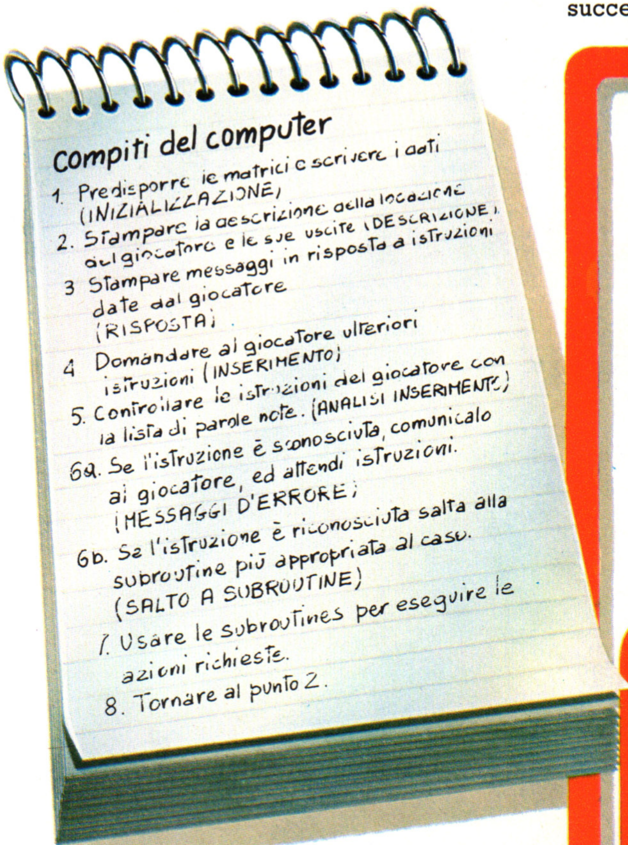
All'inizio del gioco il giocatore non ha nulla con sé; per indicare questo la matrice C( ) è vuota. Quando un oggetto viene raccolto, il computer scrive 1 nella sua casella. Quindi non servono istruzioni DATA per la matrice C( ).

Inizializzazione

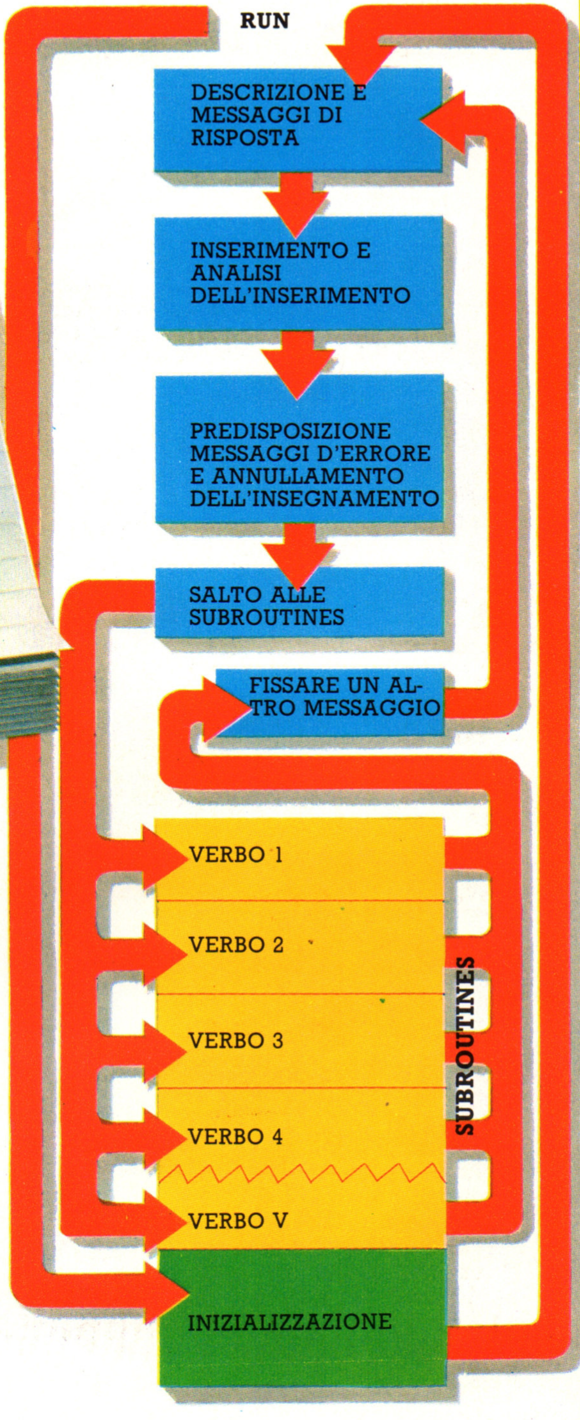
Predisporre le matrici e riempire i dati si dice "inizializzazione". Nel prossimo paragrafo vedrete dove sistemare questa operazione nella struttura del programma.

La struttura del programma

Per arrivare alla struttura globale del programma per il gioco d'avventura, dovete ancora considerare quali compiti dovranno essere eseguiti dal computer, nel corso del gioco.



La lista precedente mostra l'ordine in cui il computer deve eseguire le varie operazioni, che non è necessariamente l'ordine in cui devono apparire nel programma. Una buona parte del programma è costituito dalla routine di inizializzazione che serve una volta sola per ogni gioco, e benché questa sia la prima cosa che il computer deve fare, è buona idea metterla alla fine del programma. Questo perché, ogni volta che al computer è richiesto un GOTO o un GOSUB, questo torna all'inizio del programma e controlla ogni numero di linea finché non trova quello che gli serve. Ciò può portare via un bel po' di tempo, in un programma lungo. Ponendo la fase di inizializzazione alla fine, il computer non dovrà ricontrollarne i numeri di linea ad ogni mossa del giocatore. La struttura del programma in pratica appare come segue. Come potete vedere, la parte maggiore è costituita dalle subroutines. Infatti ci dev'essere una subroutine per ogni verbo usato nel gioco. Altre informazioni vi saranno date successivamente.



Teletext



Il teletext è un tipo di videotex che visualizza sul TV pagine di informazioni in formato schermo, prelevandole dal computer centrale. Il teletext viene diffuso sui canali TV insieme ai programmi televisivi ordinari e viene ricevuto dal vostro apparecchio TV nel modo usuale. Avete bisogno di un adattatore specifico per il teletext per decodificarne e visualizzarne i segnali digitali. Alcuni apparecchi TV hanno un decodificatore già incorporato, oppure potete acquistarne uno. Vi servirà anche una tastiera numerica per il controllo a distanza del teletext, del tipo di quella raffigurata sotto, per richiedere le pagine volute. Poiché il teletext è radioemesso, diversamente dal videodata è un sistema a senso unico. Di fatto esso differisce dal videodata per alcuni metodi che vengono illustrati in queste due pagine.

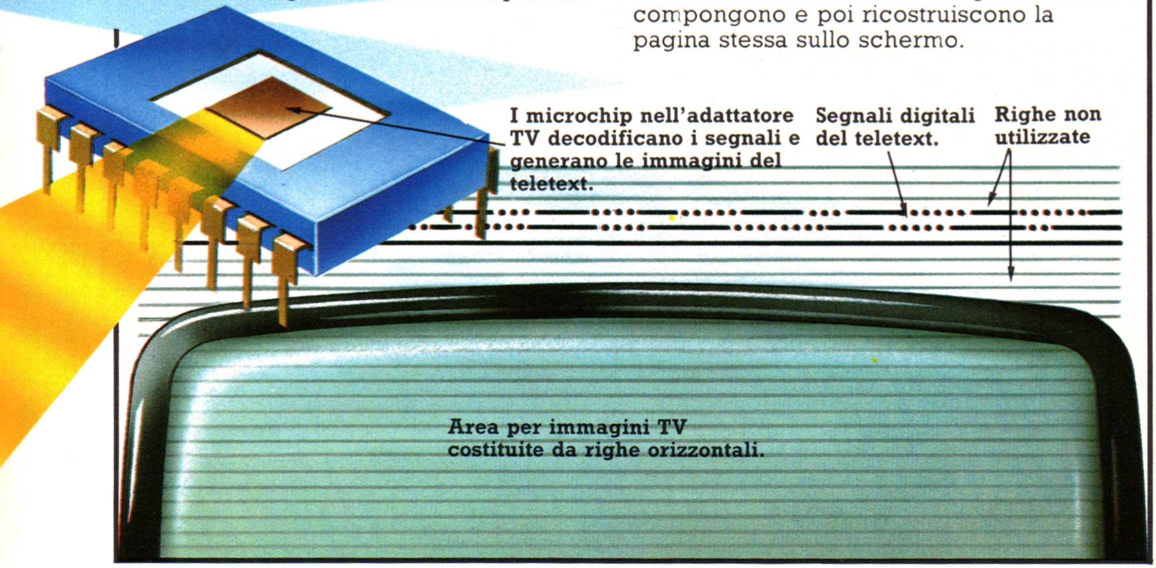
La trasmissione del teletext (TELEVIDEO)

Le pagine del teletext sono approntate e memorizzate da computer simili a quelli per il videodata. I servizi forniti dal teletext hanno generalmente meno pagine del videodata. Le pagine vengono radiodiffuse una di seguito all'altra a flusso continuo ed occorrono un paio di minuti per inviarle tutte. Quando le pagine sono state tutte trasmesse, il ciclo riprende; ciò significa che le pagine non sono tutte disponibili contemporaneamente come avviene per il videodata: dovrete attendere che venga trasmessa quella che volevate. È questione di pochi secondi. Le pagine piú comuni compaiono piú di una volta in ciascun ciclo. Le pagine sono numerate e potete scegliere quella che volete vedere mediante menu e richiamarle con la tastiera numerica. Quest'ultima, (vedi figura), ordina all'apparecchio TV di passare dai programmi ordinari al teletext. La maggior parte delle tastiere numeriche, per il controllo a distanza, funzionano inviando istruzioni codificate sotto forma di raggi infrarossi. Un ricevitore ad infrarossi inserito nel TV rileva e decodifica queste istruzioni e le passa all'adattatore per il teletext. I simboli particolari, che si vedono sulla destra, controllano le funzioni speciali che il teletext offre.



Segnali per il teletext

Tutte le immagini TV sono costituite da centinaia di linee orizzontali che variano molte volte al secondo in modo da produrre il familiare movimento delle immagini sullo schermo. Vi sono alcune linee addizionali che non vengono utilizzate per formare la normale immagine TV e sono situate nelle parti piú alta e piú bassa dello schermo. Le linee superiori sono usate per il segnali del teletext, quelle inferiori trasportano informazioni per i tecnici TV addetti ai trasmettitori. Su un apparecchio TV sintonizzato male, con l'immagine sfocata e risultando visibili le linee superiori, potreste vedere la codifica digitale dei segnali del teletext: una serie di punti luminosi in rapido movimento. Questi dicono ai microchip dell'adattatore che cosa far comparire sullo schermo per ricostruire una pagina. I microchip attendono la pagina che volete, memorizzano i segnali che la compongono e poi ricostruiscono la pagina stessa sullo schermo.



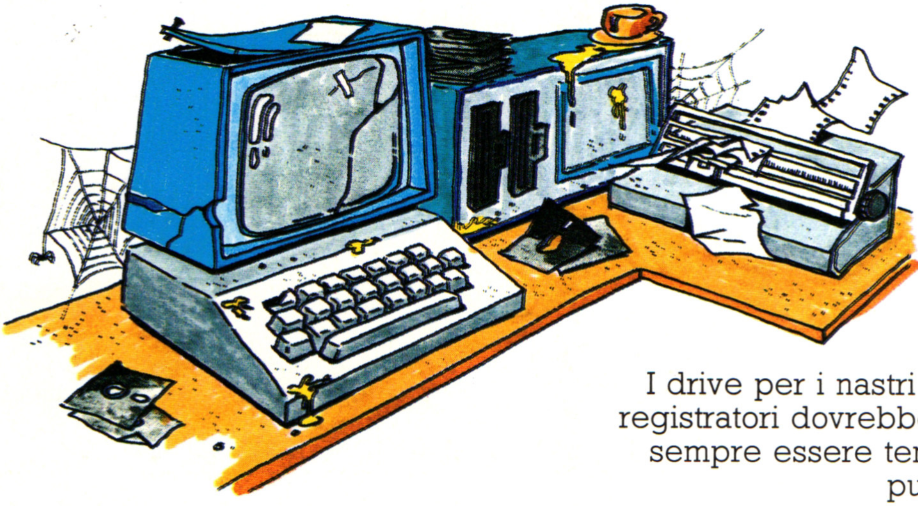
Che cosa si vede sul teletext

Come i videodata, i teletext forniscono molte informazioni utili quali notizie, risultati sportivi, situazioni meteorologiche, ricette, informazioni per chi viaggia, giochi e cosí via. Sebbene vi siano meno informazioni che sul videodata e sebbene non sia interattivo, il teletext ha alcuni vantaggi: è gratuito, se si prescinde dal costo dell'adattatore. Inoltre, poiché il teletext viene trasmesso con continuità sullo stesso



canale dei programmi televisivi e contemporaneamente ad essi, può essere sovrapposto ad un programma ordinario per fornire sottotitoli sullo schermo, unitamente alle immagini. L'adattatore può essere programmato per visualizzare una determinata pagina in un particolare momento. Potete anche chiedere di vedere una pagina appena questa è stata aggiornata, ad esempio per avere le "ultimissime" circa un fatto specifico, come raffigurato sopra.

Manutenzione dei nastri



I drive per i nastri e i registratori dovrebbero sempre essere tenuti puliti.

In particolare, le testine di registrazione e lettura andrebbero pulite, regolarmente; lo si può fare con un panno di buona qualità o un pezzetto di cotone leggermente impregnato di freon o di alcool isopropilico attaccato all'estremità di un bastoncino di plastica o di legno (non usare metalli).

Basta una piccola quantità di alcool, che non deve entrare in contatto con altre parti del meccanismo.

Come regola generale, le testine andrebbero pulite ogni 8-10 ore di utilizzazione. Controllare i consigli del fabbricante, dato che alcune testine possono richiedere solventi specifici.

Vanno puliti anche i rulli, i perni e altre guide o parti meccaniche.

I solventi devono essere usati con grande attenzione e solo secondo le istruzioni del fabbricante.

Di regola, quando si toglie una cassetta, controllare che il meccanismo non sia inquinato dalla polvere. In particolare controllare le testine e le guide, che sono i punti in cui è più probabile che si accumulino le particelle. Bordi spezzati o altri difetti del nastro provocheranno un'immediata contaminazione del meccanismo.

Quando un nastro o una cassetta sono inquinati, eliminarli. Si tratta di un problema contagioso: la materia inquinante arriverà alla testina e inquinerà altri nastri.

NOVITA' ASSOLUTA IN EDICOLA

Guida VIDEO GIOCHI

1
GIUGNO
L. 3.500

LA GRANDE GUIDA A TUTTI I GIOCHI ELETTRONICI E NON

Nuovissima, ricca e tutta a colori. GUIDA VIDEOGIOCHI ti aspetta in edicola con oltre 60 giochi recensiti, i commenti, le curiosità, i trucchi e le novità da tutto il mondo.

E, in più, partecipi al grande concorso riservato ai fedeli lettori di GUIDA VIDEOGIOCHI.

FANTASTICO CONCORSO GUIDA VIDEOGIOCHI

I premi in palio sono favolosi: due esclusive Control Deck NINTENDO e tanti game originali.



Nintendo

MATTEL

GRUPPO EDITORIALE JACKSON

Aut. Min. Rich.

CPC464 e 6128 fantastici computer, fantastici TV!

L. 399.000 + IVA



TUTTO COMPRESO.

CPC464GT 64 Kb RAM con monitor fosfori verdi, tastiera, registratore a cassetta, joystick, 100 programmi/giochi: L. 399.000.+ IVA

CPC464CTM 64 Kb RAM con monitor a colori, tastiera, registratore a cassette, joystick, 100 programmi/giochi: L. 699.000.+ IVA

CPC6128GT 128 Kb RAM con monitor a fosfori verdi, velocissimo disk driver da 3" doppia faccia (180 Kb + 180 Kb), joystick, 50 programmi/giochi: L. 699.000.+ IVA

CPC6128CTM 128 Kb RAM con monitor a colori, velocissimo disk driver da 3" doppia faccia (180 Kb + 180 Kb), joystick, 50 programmi/giochi: L. 899.000.+ IVA

WKS 6128 TV.

Stazione completa composta da: CPC 6128 CTM; Tavolo a ripiani; Sintonizzatore TV; Antenna amplificata.
Tutto a L. 999.000.+ IVA

PRONTO AMSTRAD.

Telefonaci: 02/26410511, avrai ogni informazione; oppure scrivici: Casella Postale 10794 - 20124 Milano.

LI TROVI QUI.

Presso i numerosissimi punti vendita Amstrad. Cerca quello più vicino su "Amstrad Magazine" in edicola, chiedi anche Junior Amstrad la rivita che ti regala i giochi per CPC (troverai molte notizie in più).
Oltre 150 Centri di Assistenza Tecnia.

FANTASTICO, DIVENTA TV COLOR.

Al momento del tuo acquisto puoi trasformare il tuo CPC con monitor a colori in TV color, il tuo TV color, come?
Ma è semplice, basta Acquistare il sintonizzatore TV (MP3) a L. 199.000.+ IVA



AMSTRAD

DALLA PARTE DEL CONSUMATORE

OPINIONE Comunicazione Integrata